(NeXT Tip #25b) Context Block Style

Christopher Lane (*lane[at]CAMIS.Stanford.EDU*)
Tue, 23 Mar 1993 15:24:20 -0800 (PST)

To: KSL-NeXT[at]CAMIS.Stanford.EDU
Message-ID: <MailManager.732929060.8458.lane[at]ssrg-next-1>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=US-ASCII

Those who looked closely at the 'printf:' method code in this week's 'Text Object Utilities' tip might have have noticed the odd little construct:

```
    va_start(ap, format); {
        (void) vsprintf(buffer, format, ap);
        } va_end(ap);
```

I'm not talking about the 'va_start', 'va_end' and 'vsprintf' routines, which are interesting in their own right, but the layout of the block -- this was not a formatting error. It's a personal style idiom that I use which others might find helpful. Consider this typical method from NeXT's Draw.app:

```
- speedyDraw
{
    [self lockFocus];
    NXSetColor (backgroundColor);
    NXRectFill (&rRect);
    [result composite:NX_SOVER toPoint:&rRect.origin];
    NXSetColor (NXChangeAlphaComponent (NX_COLORBLACK, 1.0));
    NXFrameRect(&rRect);
    [[self window] flushWindow];
    [self unlockFocus];


    return self;
}
```

As this type of method gets more complicated it's not always clear that some lines can only be understood in the context of lockFocus/unlockFocus and others are independent. Adding the context block style, we get:

```
- speedyDraw
{
    [self lockFocus]; {
        NXSetColor(backgroundColor);
        NXRectFill(&rRect);
        [result composite:NX_SOVER toPoint:&rRect.origin];
        NXSetColor(NXChangeAlphaComponent(NX_COLORBLACK, 1.0));
        NXFrameRect(&rRect);
        } [self unlockFocus];


    [[self window] flushWindow];


    return self;
}
```

(The 'flushWindow' call didn't need to be in the lock/unlockFocus block and moving it outside should make this clear to someone reading the program.)

The idea is that functions that setup and undo a specific context aren't very different from the right hand side of an 'if' statement and should be blocked similarly to show their relationship. There are other ways this blocking can be done, e.g. the 'do' and 'undo' calls themselves can inside the block. I personally prefer the format shown due to symmetry -- there are others that achieve the same end result. Some more NeXT-specific examples:

```
    mutex_lock(TickLock); {
        [self->tickField setIntValue:TickCount];
        } mutex_unlock(TickLock);


    [window orderFront:self]; {
        ...
        } [window orderOut:self];


    eventMask = [window removeFromEventMask:NX_MOUSEMOVEDMASK]; {
        mouse = theEvent->location;
        [window convertBaseToScreen:&mouse];
        } [window setEventMask:eventMask];
```

Being a style construct, it attempts to optimize programmer time and has no affect on runtime. (The block construct compiles into the 'flat' version.)
And like most style issues, you can always choose to take it or leave it!

- Christopher